Semester Conversion (SemCon)

Team Phoenix

George Adams          David Chung
Tony Cocco            Jacob Weigand

Project Sponsor

Registrar Office – Joe Loffredo

Faculty Coach

Robert Kuehl
# Contents

Semester Conversion (SemCon) ......................................................................................... 1

  Team Phoenix ................................................................................................................. 1

  Project Sponsor ............................................................................................................. 1

  Registrar Office – Joe Loffredo .................................................................................... 1

  Faculty Coach ............................................................................................................... 1

  Project Overview ......................................................................................................... 3

Synopsis ................................................................................................................................ 3

Purpose .............................................................................................................................. 3

Basic Requirements ........................................................................................................... 3

Creating Users .................................................................................................................... 3

Editing Users ..................................................................................................................... 4

Deleting Users ................................................................................................................... 4

Uploading COFs ................................................................................................................. 4

Editing COFs ....................................................................................................................... 4

Viewing COFs ..................................................................................................................... 5

Search for COFs ................................................................................................................... 5

Constraints .......................................................................................................................... 5

Operating Environment .................................................................................................... 5

Other Nonfunctional Requirements .................................................................................. 8

Development Process ...................................................................................................... 9

Project Team ...................................................................................................................... 9

Project Schedule: Planned and Actual ............................................................................ 10

Schedule ............................................................................................................................ 10

System Design ................................................................................................................... 13

Component Diagram ......................................................................................................... 13

Deployment Diagram ........................................................................................................ 14

Process and Product Metrics ............................................................................................. 15

Product State at Time of Delivery .................................................................................... 16

Project Reflection ............................................................................................................. 17

References .......................................................................................................................... 17
Project Overview

Synopsis
RIT is redesigning all of their academic programs to fit a new 5 year x 3 hour semester-based calendar, to begin in the fall of 2013. In preparation of this conversion every course will need to be migrated from the quarter-based calendar, and new semester courses created. All of these courses then need to be reviewed and approved at various levels of the University.

Each course will have a corresponding MS Word document that is placed in “My Courses” for review and approval. This makes it very difficult to search, share information, review or analyze the new courses. This was where our senior project team came in.

RIT asked our team to extract the data from these Course Outline Forms (COFs) and use them to populate a database. Course Outline Forms are used by RIT to manage information related to the course, i.e. any requirements it fulfills, facts about the class, goals of the course, etc. From here the COF data will be used to drive web interfaces that will facilitate searching, sorting and viewing by a variety of audiences. Once we have this data we can then track and edit it with different levels of security access. These users should be controlled and limited as to which fields are available for general access and what fields would be viewable and even updated by authorized users.

Purpose
This application will provide RIT with a way to manage COFs for the upcoming change from a quarter to semester system; this may include uploading, editing, and viewing COF information on a website. It will be available to the entire RIT community.

The application may possibly be used to populate course information in the new student information system that will roll out when the conversion to a semester system is complete.

Basic Requirements

In order to understand the basics of the system it is important to note user roles that currently exist: SuperUser, College Admin, College Editor, and Public. A SuperUser is most likely anyone from the registrar's office or a scheduling officer who has access over every feature in the system, most notably: manage departments and create other users. College Admins are heads of the individual colleges and are responsible for creating college editors and uploading courses. College Editors, users designated by College Admins or SuperUsers, can only edit courses for the college they have been assigned. Public users only have access to the bare minimum. They can browse and search the course library.

Creating Users

Users
Only SuperUsers and College Admins can use this feature.

Input
User can create a new user by supplying a RIT Account, Role, and College.

*Output*
The new user is added to the database.

**Editing Users**

*Users*
Only SuperUsers and College Admins can use this feature.

*Input*
User selects a user to edit and changes the user’s RIT Account, Role, and/or College.

*Output*
The user’s information is modified in the database.

**Deleting Users**

*Users*
Only SuperUsers and College Admins can use this feature.

*Input*
User can create a new user by supplying a RIT Account, Role, and College.

*Output*
The user is deleted from the database.

**Uploading COFs**

*Users*
Only SuperUsers and College Admins can use this feature.

*Input*
User chooses a .doc file or zip file containing multiple .doc files to upload to the application.

*Output*
Information from each .doc file is put into the database.

**Editing COFs**

*Users*
Only SuperUsers, College Admins, and College Editors can use this feature.

*Input*
User selects a COF to edit and changes fields within the COF.

*Output*
The changes made to the COF are updated in the database.
Viewing COFs

*Users*
All users can use this feature.

*Input*
User selects a COF through the course browser or search.

*Output*
The select COF’s information is retrieved from the database and displayed.

Search for COFs

*Users*
All users can use this feature.

*Input*
The user provides input for the search parameters: college, department, title, or replaces/equivalent to.

*Output*
Any records matching the search criteria from the database are retrieved and displayed.

Constraints

Many of the constraints listed below are based the ITS departments and procedures.

Operating Environment

**Client side**
The application can be run on any operating system. The computer used must have an internet connection. The application must run on major web browsers Internet Explorer 7+, Mozilla Firefox 3+, Safari 5+, and Google Chrome 8+ supporting javascript.

**Server side**
The server will be using CentOS 5.5 linux operating system; the application will run on Tomcat 5.5.27 (servlet 2.4, jsp 2.0) within Apache 2.2.16.

Design and Implementation Constraints

**Database**

Oracle Version: 11.2.0.1 Enterprise Edition with no individually priced optional features in use except Advanced Security Option (ASO) for encryption (no use of RAC, Partitioning, Spatial, etc)

General Database Design Standards:
Instance names are limited to 8 characters, the first 4 reflect the application name & the last 4 designate the function (DEVL, TEST or PROD).

The first 4 characters will be “SEMC”.

- **semc_owner** - owns all the database objects.
- **semc_user** - full access to {functional name}_owner used for app server access.
- **semc_read** - read only access for app server if required.

At the highest level, a login should support only one application or application server unless the application server is clustered or load balanced. This enables clearer visibility on when & where access is coming from.

- **SEMC_DATA** - tablespace to build tables for application data.
- **SEMC_INDEX** - tablespace to build indices on application tables.

**DBA Services:**

ITS DBA is responsible for the physical well being of the database. The application developer is responsible for the integrity of the customer data. To that end, the application developer does not have privileges to construct physical files or change system level settings. They will however have full access to the logins outlined above.

**Web Application**

The web application uses the Struts2 framework (version 2.0.14), and Oracle Database 11g Release 2 (11.2.0.1) JDBC Drivers. For persistence, Oracle 11g Release 2 (11.2.0.1) is used. The application server and the database are on separate machines. The application server runs on CentOS 5.5. The application will be developed using Java 1.6 as the programming language using servlet 2.4 and jsp 2.0 specifications.

**Deployment standards**

The application should be in a .war file (semcon.war)

Properties should be modal and contained within the app. Mode is set in the environment by a file contained in /home/webapps/properties/semcon.properties. The mode file contains a single line ( e.g. mode=production).

Deployments are made by doing a build on a build machine and then copying the war to the dev environment. Once QA and change management has approved the version in dev the war file will be moved to the test environment. Moves to production happen through the same process.
Login authentication

Shibboleth will be used for user login.

External Interface Requirements

User Interfaces

The user interface must account for color-blind individuals. Usability testing will be used to determine a suitable interface for most users (time permitting).

Hardware Interfaces

In development, the application will be run on our SE provided server and use an Oracle database provide by ITS. In test the application will be deployed on an ITS provided server. In production the application will be deployed on multiple nodes and load balanced.

Software Interfaces

Production Web Environment
Apache 2.2.16
Tomcat 5.5.27 (servlet 2.4, jsp 2.0)
Java 1.6
CentOS 5.5 – Operating system
note: web applications are deployed across multiple nodes and load balanced

Development Web Environment
Apache 2.2.14
Tomcat 5.5.27 (servlet 2.4, jsp 2.0)
Java 1.6
Ubuntu 10.04 LTS – Operating system

Libraries

struts 2.2.1

jquery 1.4.x

ojdbc14.jar (datasource will be defined in the server.xml of the container)

Build tools

ant1.8.x (there should be ant tasks to build the .war, build file should be located in a build folder in the project)

Database tools
SQL Developer

Communications Interfaces

The application will deployed across multiple nodes and load balanced by ITS. SuperUsers, College Admins, and College Editors login screen is encrypted, as well as their entire session.

Other Nonfunctional Requirements

Performance Requirements

The application will perform to standards set by the ITS department at RIT. Searching and viewing COFs should be quick, within a minute for searching. When COFs are uploaded to the application, each COF should be parsed within one second.

Safety Requirements

The application must be self contained and should not affect outside systems and databases.

Security Requirements

Only users with the proper permissions can modify COF data. Login will use ITS provided Shibboleth system over https.

Software Quality Attributes

The quality attributes of main concern are correctness, availability, and robustness. Robustness deals with how the system will handle errors in the COFs submitted. Availability of the application should be 100% of business hours as defined as 8a-10p Monday through Friday.
The design process prescribed is an iterative model. This process changed from a modified waterfall with the introduction of ITS department’s process overhead. Technical requirements and demands from ITS department as development progressed was re-evaluated and reviewed after each milestone. Development and test produced a release that was submitted to end users for testing. Requirements are then re-evaluated when necessary.

Development and test phases occurred iteratively as proposed by the milestone schedule; iterations consisted of new functionality and respective test suites. After the final iteration, the system was required to pass full acceptance testing before it was officially released to production by the ITS department.

**Project Team**

<table>
<thead>
<tr>
<th>Role</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team Lead</td>
<td>Tony Cocco</td>
</tr>
<tr>
<td>Quality Assurance</td>
<td>Tony Cocco</td>
</tr>
<tr>
<td>Sponsor Contact</td>
<td>Jacob Weigand</td>
</tr>
<tr>
<td>Release Builder</td>
<td>George Adams</td>
</tr>
<tr>
<td>Webmaster</td>
<td>George Adams</td>
</tr>
<tr>
<td>System Admin</td>
<td>George Adams</td>
</tr>
<tr>
<td>Database Admin</td>
<td>David Chung</td>
</tr>
</tbody>
</table>
The team decided roles by past experience and desire to working with certain technologies.

**Project Schedule: Planned and Actual**

<table>
<thead>
<tr>
<th>Milestones</th>
<th>Description</th>
<th>Milestone Criteria</th>
<th>Planned Date</th>
<th>Actual Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>M0</td>
<td>Requirements</td>
<td></td>
<td>2010-11-30</td>
<td>2010-11-30</td>
</tr>
<tr>
<td></td>
<td>Meet with faculty mentor and sponsor</td>
<td>Attend expectations presentation, meet with sponsor, and meet with faculty mentor, Requirements gathering</td>
<td>2010-12-02</td>
<td>2010-12-02</td>
</tr>
<tr>
<td>M1</td>
<td>Planning</td>
<td></td>
<td>2010-12-02</td>
<td>2010-12-02</td>
</tr>
<tr>
<td></td>
<td>Complete initial documentation for the project</td>
<td>Finalized Project Plan, SRS, Process description, VM setup, site setup, Redmine setup, develop test plan, migration process, architecture diagram, database diagram, design document</td>
<td>2010-12-16</td>
<td>2010-12-16</td>
</tr>
<tr>
<td>M2</td>
<td>Design</td>
<td></td>
<td>2010-12-16</td>
<td>2010-12-16</td>
</tr>
<tr>
<td></td>
<td>Create system architecture</td>
<td>Design agreed upon and submitted to faculty mentor for validation</td>
<td>2011-01-04</td>
<td>2011-01-04</td>
</tr>
<tr>
<td>M3</td>
<td>Development Iteration (Fall)</td>
<td></td>
<td>2011-01-04</td>
<td>2011-01-04</td>
</tr>
<tr>
<td>M3.1 “Adarna”</td>
<td>database schema database installed with fixture of a couple hand-entered documents basic search</td>
<td>2011-01-18</td>
<td>2011-01-18</td>
<td></td>
</tr>
<tr>
<td>M3.2</td>
<td>&quot;Bennu&quot;</td>
<td>web view of documents</td>
<td>2011-02-01</td>
<td>2011-02-01</td>
</tr>
<tr>
<td>M3.2.1</td>
<td>Deploy to ITS Dev</td>
<td>Intermediate release of Bennu to ITS dev to work on migration process and begin user testing.</td>
<td>2011-03-17</td>
<td>2011-03-17</td>
</tr>
<tr>
<td>M4</td>
<td>Interim Presentation</td>
<td>2011-02-11</td>
<td>2011-02-11</td>
<td></td>
</tr>
<tr>
<td>M5</td>
<td>Development Iteration (Spring)</td>
<td>2011-02-21</td>
<td>2011-02-21</td>
<td></td>
</tr>
<tr>
<td>M5.1</td>
<td>&quot;Garuda&quot;</td>
<td>Shibboleth login User roles established along with access restrictions Additional search features(SE/SR, alpha code)</td>
<td>2011-03-15</td>
<td>2011-03-15</td>
</tr>
<tr>
<td>M5.2</td>
<td>&quot;Roc_RC1&quot;</td>
<td>UI adjustments Final tweaking of functionality Re-validate parser</td>
<td>2011-04-05</td>
<td>2011-04-05</td>
</tr>
<tr>
<td>M5.2a</td>
<td>&quot;Roc_RC2&quot;</td>
<td>Updates from M6.1.</td>
<td>2011-04-19</td>
<td>2011-04-19</td>
</tr>
<tr>
<td>M5.2b</td>
<td>&quot;Roc_RC3&quot;</td>
<td>General updates and changes.</td>
<td>2011-05-03</td>
<td>2011-05-03</td>
</tr>
<tr>
<td>M5.3</td>
<td>&quot;Roc&quot;</td>
<td>Final release</td>
<td>2011-05-10</td>
<td>2011-05-04</td>
</tr>
<tr>
<td>M6</td>
<td>Acceptance</td>
<td>2011-04-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M6.1</td>
<td>Testing performed on SIS temp</td>
<td>All tests are re-run and should pass.</td>
<td>2011-04-14</td>
<td>2011-04-28</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------</td>
<td>--------------------------------------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>M6.2</td>
<td>Testing performed on SIS test</td>
<td>Testing is performed again in the SIS test environment.</td>
<td>2011-04-21</td>
<td>2011-05-06</td>
</tr>
<tr>
<td>M7</td>
<td>Final Project + Presentation</td>
<td></td>
<td>2011-04-14</td>
<td>2011-04-14</td>
</tr>
<tr>
<td>M7.1</td>
<td></td>
<td>Create final project poster and present to IAB.</td>
<td>2011-04-29</td>
<td>2011-04-29</td>
</tr>
<tr>
<td>M7.2</td>
<td></td>
<td>Final presentation.</td>
<td>2011-05-12</td>
<td>2011-05-12</td>
</tr>
<tr>
<td>M8</td>
<td>Release Product</td>
<td></td>
<td>2011-05-02</td>
<td>2011-05-02</td>
</tr>
<tr>
<td></td>
<td>Deploy on ITS production environment</td>
<td>Project is deployed and hosted on ITS server space and working properly. Also deliver any training material and additional documents for ITS.</td>
<td>2011-05-06</td>
<td>2011-05-09</td>
</tr>
<tr>
<td>M9</td>
<td>Close Project</td>
<td></td>
<td>2011-05-21</td>
<td>2011-05-21</td>
</tr>
</tbody>
</table>
System Design

Component Diagram

The component diagram above follows the structure of Struts 2 where the JSPs act as the View, Action classes as the Controller, and Model as the Model. Primary feature is uploading COFs to the application.

Here is an sequence of actions when uploading a course:

1. User accesses the Upload JSP and uploads a COF doc file.
2. Before UploadAction is executed, Validators are run to check user permissions.
3. UploadAction executes, which passes the COF to the Parser.
4. Parser reads the COF and puts the data into a CourseModel object.
5. The CourseModel object data is validated by Problem.
6. The CourseModel object is returned to UploadAction.
7. UploadAction calls method in CourseModel to save the COF in the database.
8. UploadAction returns the Results JSP.

**Deployment Diagram**

The deployment structure was predetermined by the ITS department as this what currently used for Java web applications. Above the application deployed to multiple Tomcat servers where load on each server is handled by the Apache2 Web Server. Each Tomcat server hosting our application is connected two components, the Oracle 11g Database and Shibboleth. The Oracle 11g Database contains all data managed within our application. Shibboleth handles login authentication into our application.
The diagram above shows how each feature is processed by our system when performed by the users seen at the top. At the top, Users use a feature through JSPs which works with the appropriate action class in the middle. The each action class works with model classes which handle data management. In the case of the COF model class, it works with parser to extract data from COF doc files.

**Process and Product Metrics**

Three metrics were collected through the project. The following are the values of each metric as collected per-milestone. Metrics listed as NA were not collected as they did not relate to the milestone or exist. Notably, time spent in M1 was only 14 days but due to ITS requirements and revisions became a living and persistent cycle, making an exact count worthless. Metrics provided us with little surprising information. Defect count rose as we tackled more technical issues and received user feedback, which is to be expected. Time spent per milestone varied with functionality assigned to that milestone, synonymous with velocity in agile practices.

**Metrics**

<table>
<thead>
<tr>
<th>Type of data</th>
<th>Purpose</th>
<th>Responsible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of data</td>
<td>Purpose</td>
<td>Responsible</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Defects injected per phase</td>
<td>Identify problems with the development process iterations</td>
<td>Quality Assurance</td>
</tr>
<tr>
<td>Time spent per phase</td>
<td>Team total time spent per iteration in order to assess work balance</td>
<td>Quality Assurance</td>
</tr>
<tr>
<td>Average time to parse course outline form</td>
<td>Identify speed at which system can process document</td>
<td>Quality Assurance</td>
</tr>
</tbody>
</table>

Metrics were collected at the end of each iteration. Iterations are ended when designated features have been completed and verified.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Defects found</th>
<th>Time spent (days)</th>
<th>Average time to parse course outline form (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M0</td>
<td>0</td>
<td>3</td>
<td>NA</td>
</tr>
<tr>
<td>M1</td>
<td>0</td>
<td>NA(14)</td>
<td>NA</td>
</tr>
<tr>
<td>M2</td>
<td>2</td>
<td>19</td>
<td>NA</td>
</tr>
<tr>
<td>M3.1</td>
<td>4</td>
<td>14</td>
<td>0.13</td>
</tr>
<tr>
<td>M3.2</td>
<td>3</td>
<td>14</td>
<td>&lt;0.1 (warm)</td>
</tr>
<tr>
<td>M4</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>M5.1</td>
<td>8</td>
<td>27</td>
<td>&lt;0.1 (warm)</td>
</tr>
<tr>
<td>M5.2</td>
<td>11</td>
<td>32</td>
<td>&lt;0.2</td>
</tr>
<tr>
<td>M5.2a</td>
<td>7</td>
<td>14</td>
<td>&lt;0.2</td>
</tr>
<tr>
<td>M5.2X</td>
<td>2</td>
<td>14</td>
<td>&lt;0.2</td>
</tr>
<tr>
<td>M6*</td>
<td>NA</td>
<td>21</td>
<td>NA</td>
</tr>
</tbody>
</table>

*Incorporated within M5.2x

Product State at Time of Delivery

The application is currently feature complete and is in final acceptance testing before being deployed onto production by the ITS department.

Initial project planned features missing

Download COF as PDF file
This feature was removed as the team thought the amount of effort required to create PDF file was not worth value. It would not be frequently used as COFs may be viewed online in a similar format to COF docs.

**Unplanned features added**

*Add/Edit/Delete Departments*

This feature was added to improve the maintainability of the application. It provided the Registrar’s office the ability to add new Department as they are being made for the Semester Conversion project easily. It also validated Department fields in uploaded COFs and referenced when performing searches on COFs in certain Departments.

*Sections exclusive to NTID COFs*

NTID COFs had critical sections that did not follow the standard COF. A new field was added to NTID COFs and additional information was added the “Relevant Information” section seen in standard COFs.

**Project Reflection**

The project was successful and has been finished to completion. A few items that the team thought that went well included delivering the product, support from ITS and our Sponsor, and passing the project to ITS. We delivered an application that has satisfied our Sponsor’s and ITS’s requirements. The team had great success on getting feedback and assistance on the project from both the ITS department and our Sponsor. On passing the project to ITS, they were happy with the documentation we have provided on maintaining our application.

As with all projects there are always improvements that can be made. Throughout the project time conflicts proved to hinder our work; job and work on other class often was prioritized over our project work. Another issue we had was a lack of formal test procedures. We noticed this when ITS asked our team to deliver formal test procedures in the first half of the project. Some quality features were put aside in our application because the short-life expectancy; this was considered because major future work was not expected.

Moving forward in the future, the team would increase team communication to deal with time conflicts by scheduling more efficiently.

**References**